Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. – 10. (Cancelled)

11. (Original) An error control method for multilevel memory cells operating with a variable number of storage levels, the method comprising:

receiving a first information word having k input symbols each in a first base;

converting the first information word into a second base by converting the input symbols into input symbols in the second base;

encoding the converted first information word into a first codeword having k + n coded symbols in the second base;

writing the first codeword into the multilevel memory cells.

12. (Original) The error control method of claim 11, further comprising:

receiving a second information word having k input symbols each in the second base;

encoding the second information word into a second codeword having k + n coded symbols in the second base;

writing the second codeword into the multilevel memory cells.

13. (Original) The error control method of claim 11, further comprising:

reading from the multilevel memory cells the first codeword;

decoding the first codeword into an estimated word having k estimated symbols in the second base; and

converting the estimated word into the first base by converting the estimated symbols into estimated symbols in the first base.

14. (Original) An error control method for multilevel memory cells operating with a variable number of storage levels, the method comprising:

reading from the multilevel memory cells a codeword having k + n coded symbols in a first base;

decoding the codeword into an estimated word having k estimated symbols in the first base; and

converting the estimated word into a second base by converting the estimated symbols into estimated symbols in the second base.

15. (Currently Amended) A computer storage device, comprising:

a memory matrix that includes multilevel memory cells capable of storing data in a first base or a second base;

an input transcoder having a word input that receives an information word having k input symbols each in the first base, a control input that receives a control signal indicating whether the memory matrix is operating according to the first base or the second base, and an output that outputs a converted information word in the second base;

an encoder coupled to the output of the input transcoder and structured to encode the converted information word into a codeword having k + n coded symbols in the second base;

a write circuit having an input coupled to the encoder and an output coupled to the memory matrix, the write circuit being structured to ~~writ~~ write the codeword into the memory matrix.

16. (Currently Amended) The method of claim 11 wherein the storage levels of the memory cells can assume values of the set $\{b^{a_1}, b^{a_1 a_2}, \ldots, b^{a_1 a_2 \ldots a_h}\}$, wherein $b^{a_1}, b^{a_1 a_2}, \ldots, b^{a_1 a_2 \ldots a_h}$ are bases expressed a base value b and polynomial degrees $a_1, a_2, \ldots, a_h$, the first information word is formed by k q-ary symbols belonging to an alphabet containing q different

symbols, with $q \in \{b^{a_1}, b^{a_1 a_2}, \ldots, b^{a_1 a_2 \cdots a_h}\}$, the first codeword is formed by k + n q-ary symbols, with $q=b^{a_1 a_2 \cdots a_h}$, and having an error-correction capacity t, and the encoding step includes generating through an operation of multiplication between the first information word and a generating matrix, the ~~method~~ encoding step further comprising:

acquiring values of k, t, b^{a_1}, $b^{a_1 a_2}$, …, $b^{a_1 a_2 \cdots a_h}$, which constitute design specifications of said first codeword;

calculating, as a function of $q=b^{a_1}$, k and t, the minimum value of n such that a Hamming limit is satisfied;

calculating the maximum values $\hat{n}$ and $\hat{k}$ of n and k that satisfy said Hamming limit for $q=b^{a_1}$, t and $(\hat{n}-\hat{k})=(n-k)$;

determining, as a function of t, the generating matrix of the first codeword $(\hat{n}, \hat{k})$ on a finite-element field $GF(b^{a_1})$;

constructing binary polynomial representations of finite-element fields $GF(b^{a_1})$, $GF(b^{a_1 a_2})$, …, $GF(b^{a_1 a_2 \cdots a_h})$;

identifying, using said ~~exponential~~ binary polynomial representations, elements of the finite-element field $GF(b^{a_1 a_2 \cdots a_h})$ isomorphic to elements of the finite-element fields $GF(b^{a_1})$, $GF(b^{a_1 a_2})$, …, $GF(b^{a_1 a_2 \cdots a_{h-1}})$);

establishing biunique correspondences between the elements of the finite-element fields $GF(b^{a_1})$, $GF(b^{a_1 a_2})$, …, $GF(b^{a_1 a_2 \cdots a_{h-1}})$ and the elements of the finite-element field $GF(b^{a_1 a_2 \cdots a_h})$ that are isomorphic to them; and

substituting each of a plurality of elements of said generating matrix with a corresponding isomorphic element of the finite-element field $GF(b^{a_1 a_2 \cdots a_h})$, thus obtaining a multipurpose generating matrix defining, together with said biunique correspondences, said first codeword.

17. (Previously Presented) The method of claim 16, in which said first codeword is a linear block code.

18. (Previously Presented) The method of claim 16, in which the identification of the elements of the finite-element field $GF(b^{a_1 a_2 \cdots a_h})$ isomorphic to the elements of the finite-element fields $GF(b^{a_1})$, $GF(b^{a_1 a_2})$, ..., $GF(b^{a_1 a_2 \cdots a_{h-1}})$ is performed on the basis of the multiplicity of said elements in the finite-element field $GF(b^{a_1 a_2 \cdots a_h})$.

19. (Currently Amended) The method of claim 18, in which said step of identifying the elements of the finite-element field $GF(b^{a_1 a_2 \cdots a_h})$ isomorphic to the elements of the finite-element fields $GF(b^{a_1})$, $GF(b^{a_1 a_2})$, ..., $GF(b^{a_1 a_2 \cdots a_{h-1}})$ comprises the step of identifying the elements of the finite-element field $GF(b^{a_1 a_2 \cdots a_h})$ that have respectively multiplicity being equal to, or being a function of, b^{a_1}, $b^{a_1 a_2}$, ..., $b^{a_1 a_2 \cdots a_{h-1}}$.

20. (Previously Presented) The method of claim 16, in which said step of establishing biunique correspondences between the elements of the finite-element fields $GF(b^{a_1})$, $GF(b^{a_1 a_2})$, ..., $GF(b^{a_1 a_2 \cdots a_{h-1}})$ and the elements of the finite-element field $GF(b^{a_1 a_2 \cdots a_h})$ isomorphic to them comprises the steps of:

constructing binary polynomial representations of the finite-element fields $GF(b^{a_1})$, $GF(b^{a_1 a_2})$, ..., $GF(b^{a_1 a_2 \cdots a_h})$; and

establishing biunique correspondences between the elements of the finite-element fields $GF(b^{a_1})$, $GF(b^{a_1 a_2})$, ..., $GF(b^{a_1 a_2 \cdots a_{h-1}})$ and the elements of the finite-element field $GF(b^{a_1 a_2 \cdots a_h})$ isomorphic to them, using said binary polynomial representations.

21. (Previously Presented) The method of claim 20, in which said binary polynomial representations of the finite-element fields $GF(b^{a_1})$, $GF(b^{a_1 a_2})$, ..., $GF(b^{a_1 a_2 \cdots a_h})$ are constructed using respective primitive polynomials of a degree respectively equal to a_1, $a_1 a_2$, ..., $a_1 a_2 \ldots a_h$ on the finite-element field $GF(2)$, said binary polynomial representations associating, to each element of the respective finite-element fields $GF(b^{a_1})$, $GF(b^{a_1 a_2})$, ..., $GF(b^{a_1 a_2 \cdots a_h})$, a corresponding binary polynomial of degree respectively less than a_1, $a_1 a_2$, ..., $a_1 a_2 \ldots a_h$.

22. (Previously Presented) The method of claim 21, in which said step of establishing biunique correspondences between the elements of the finite-element field $GF(b^{a_1a_2\cdots a_h})$ and the elements of the finite-element fields $GF(b^{a_1})$, $GF(b^{a_1a_2})$, …, $GF(b^{a_1a_2\cdots a_{h-1}})$ isomorphic to them comprises the steps of:

for each element of the finite-element field $GF(b^{a_1a_2\cdots a_h})$ isomorphic with a corresponding element of one of the finite-element fields $GF(b^{a_1})$, $GF(b^{a_1a_2})$, …, $GF(b^{a_1a_2\cdots a_{h-1}})$, forming a first binary word with the coefficients of the terms of the binary polynomial associated to said element of $GF(b^{a_1a_2\cdots a_h})$, and a second binary word with the coefficients of the terms of the binary polynomial associated to said corresponding element of one of the finite-element fields $GF(b^{a_1})$, $GF(b^{a_1a_2})$, …, $GF(b^{a_1a_2\cdots a_{h-1}})$;

converting said first binary word into the base $b^{a1a2\ldots ah}$, thus obtaining a first symbol, and said second binary word into the base of the finite-element field to which said corresponding element belongs, thus obtaining a second symbol; said first symbol and said second symbol defining the biunique correspondence between said element of the finite-element field $GF(b^{a_1a_2\cdots a_h}$ and the corresponding element of one of the finite-element fields $GF(b^{a_1})$, $GF(b^{a_1a_2})$, …, $GF(b^{a_1a_2\cdots a_{h-1}})$ isomorphic to it.

23. (Previously Presented) The method of claim 16, comprising:

determining, for each element of the multipurpose generating matrix, the number of logic gates necessary for carrying out the operation of multiplication associated with said element; and

abbreviating said multipurpose generating matrix to the value of k initially specified, thus minimizing the number of elements of the finite-element field $GF(b^{a_1a_2\cdots a_h})$ that require the largest number of logic gates.

24. (Previously Presented) The method of claim 23, in which said step of determining the number of logic gates is performed using said binary polynomial representation of the finite-element field $GF(b^{a_1a_2\cdots a_h})$.

25. (Previously Presented) The method of claim 11 wherein the encoding step includes generating the first codeword through an operation of multiplication between the first information word and a generating matrix, the first base is equal to a number of storage levels at which said memory cells operate, and the second base is equal to the maximum number of storage levels of said memory cells, the method further comprising:

decoding a word read in said memory, using said first codeword; and

converting, symbol by symbol, said decoded word from the second base to the first base.

26. (Previously Presented) A computer storage device, comprising:

a memory matrix of multilevel memory cells operating with a variable number of storage levels;

means for receiving a first information word having k input symbols each in a first base;

means for converting the first information word into a second base by converting the input symbols into input symbols in the second base;

means for encoding the converted first information word into a first codeword having k + n coded symbols in the second base;

means for writing the first codeword into the multilevel memory cells.

27. (Previously Presented) The device of claim 26, further comprising:

means for receiving a second information word having k input symbols each in the second base; wherein the encoding means includes means for encoding the second information word into a second codeword having k + n coded symbols in the second base, and the writing means includes means for writing the second codeword into the multilevel memory cells.

28. (Currently Amended) The device of claim 26, further comprising:

means for reading from the multilevel memory cells the first codeword;

means for decoding the first codeword into an estimated word having k estimated symbols in the second base; and

means for converting the estimated word into the first base by converting the estimated symbols into estimated symbols in the first base.

29. (New) The method of claim 14 wherein the first base is equal to a maximum number of storage levels of said memory cells, and the second base is equal to a number of storage levels at which said memory cells operate.

30. (New) The method of claim 14, the method further comprising:

converting an information word, having k input symbols each in the second base, into the first base by converting the input symbols into input symbols in the first base; and

generating the codeword by multiplying a generating matrix by the information word in the first base; and

writing the codeword into the multilevel memory cells.

31. (New) The computer storage device of claim 15, further comprising:

a read circuit coupled to the memory matrix and structured to read the codeword from the memory matrix;

a decoder coupled to the read circuit and structured to decode the codeword into an estimated word having k estimated symbols in the second base; and

an output transcoder coupled to the decoder and structured to convert the estimated word into the first base by converting the estimated symbols into estimated symbols in the first base.